Remarks

Status of the Claims

Claims 1-55 were pending in the application. In the Office Action mailed March 1, 2007, claims 1-16, 37-52, 54, and 55 were rejected. Claims 17-36 and 53 remain withdrawn from consideration as being directed to a non-elected species. By this paper, claims 1, 15, 37-41, and 45-52 have been amended. Claims 6-8, 42-44, 54, and 55 have been canceled without prejudice or disclaimer. For the reasons set forth below, Applicant submits that each of the pending claims is patentably distinct from the cited prior art and in condition for allowance. Reconsideration of the claims in view of the amendments and following remarks is therefore respectfully requested.

Claim Objections

Claims 38 and 40-51 were objected to because they depend from a withdrawn claim. Applicant has amended the affected claims to correct this informality.

Claim Rejections

Claims 1-13, 16, 37-49, and 52 were rejected under 35 U.S.C. 102(e) as being allegedly anticipated by Bruck et al. ("Bruck"). Claims 14-15, 50-51, and 54-55 were rejected under 35 U.S.C. 103(a) as being unpatentable over Bruck. These rejections are respectfully traversed. As set forth below, Applicant respectfully submits that each of the pending claims, as amended, is patentably distinct from the cited prior art.

As amended, claim 1 recites a method for providing supplemental information related to a television program, comprising:

storing a <u>plurality of program interface objects</u> (PIOs) within a memory of an entertainment system, each PIO comprising a <u>single discrete data structure</u> within the memory that encapsulates:

<u>attribute data</u> for one or more attributes providing information about a single television program, at least one attribute comprising a link to supplemental information related to the television program,

<u>program code</u> for one or more user-selectable actions executable by the entertainment system in connection with the same television program, wherein the program code comprises a <u>routine in a machine independent format</u> that is executable in a <u>Java virtual machine</u> within the interactive television system and any destination device to which the PIO is sent such that the routine does not need to be installed on the destination device prior to receiving the PIO in order to perform the associated user-selected action; and

<u>graphical data for display as an icon to visually represent the PIO within a graphical user interface</u> (GUI), <u>the GUI being other than a grid-based electronic program guide (EPG) including rows corresponding to television channels</u>,

<u>displaying the icons representing the plurality of PIOs in the GUI</u>;

in response to a user picking a icon of a selected PIO within the GUI, displaying a <u>context menu listing the one or more actions associated with the selected PIO</u>, at least one of the actions for displaying supplemental information related to the associated television program;

in response to the action for displaying supplemental information being selected from the context menu, using the at least one attribute comprising the link to retrieve the supplemental information referenced by the link; and

displaying the supplemental information in the GUI.

Support for these amendments may be found, for example, in FIGs. 7, 8, and 21, and their accompanying text.

17

A PIO, as claimed, is a discrete (separate) data structure within a memory of an entertainment system for representing a single television program. The claimed PIO not only stores attribute data for providing information about the television program (*e.g.*, title, description, rating, running time), but also includes actions comprising program code (routines), which are executable by the entertainment system in connection with the same television program (*e.g.*, program code for causing the entertainment system to record the program). The PIO further includes graphical data for an icon to represent the PIO within a non-traditional electronic program guide (EPG), as described in greater detail below. Finally, all of the claimed attribute data, program code, and graphical data are separately "encapsulated" within a single data structure, meaning that they are "encase[d] in or as if in a capsule" or that the PIO "form[s] a capsule or sheath around" the actions or attributes. See www.dictionary.com.
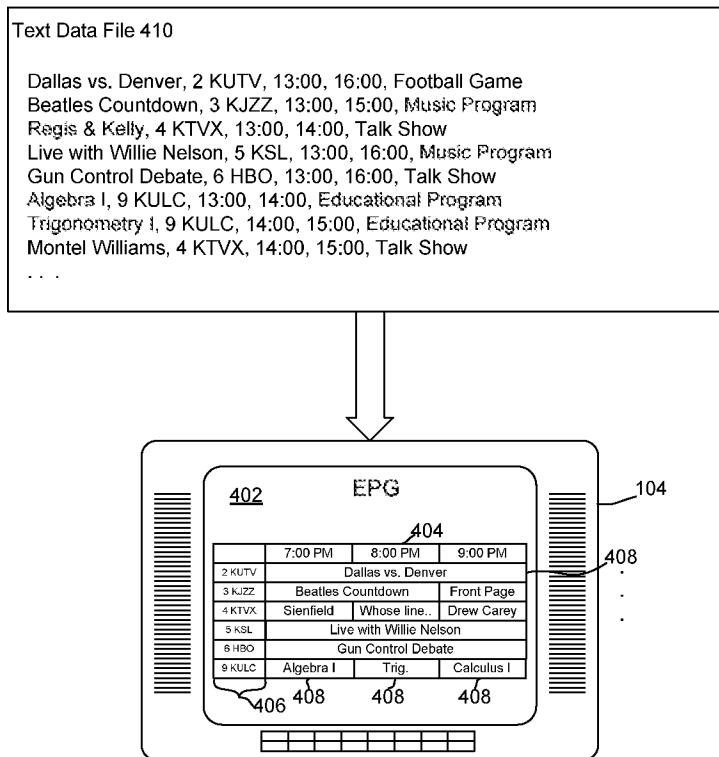
As illustrated below in Figure 7 of the present application, each PIO is a separate data structure including the necessary attribute data and program code for interacting with the corresponding single television program. For instance, the PIO 502a includes a number of attributes, such as the name of the program ("Dallas v. Denver"), the starting time ("Monday, 7:00 p.m."), etc. In addition, the PIO 502a includes a number of actions ("Record"), which are represented within the PIO 502a as program code (routines) for carrying out the actions. Thus, the actions are not merely icons or names displayed in a GUI, or even HTML links. Each set of attribute data, program code, and graphical data for a given television program is encapsulated within a single discrete PIO data structure (object), making it easier to transmit, copy, or otherwise manipulate the data as a unit.

18
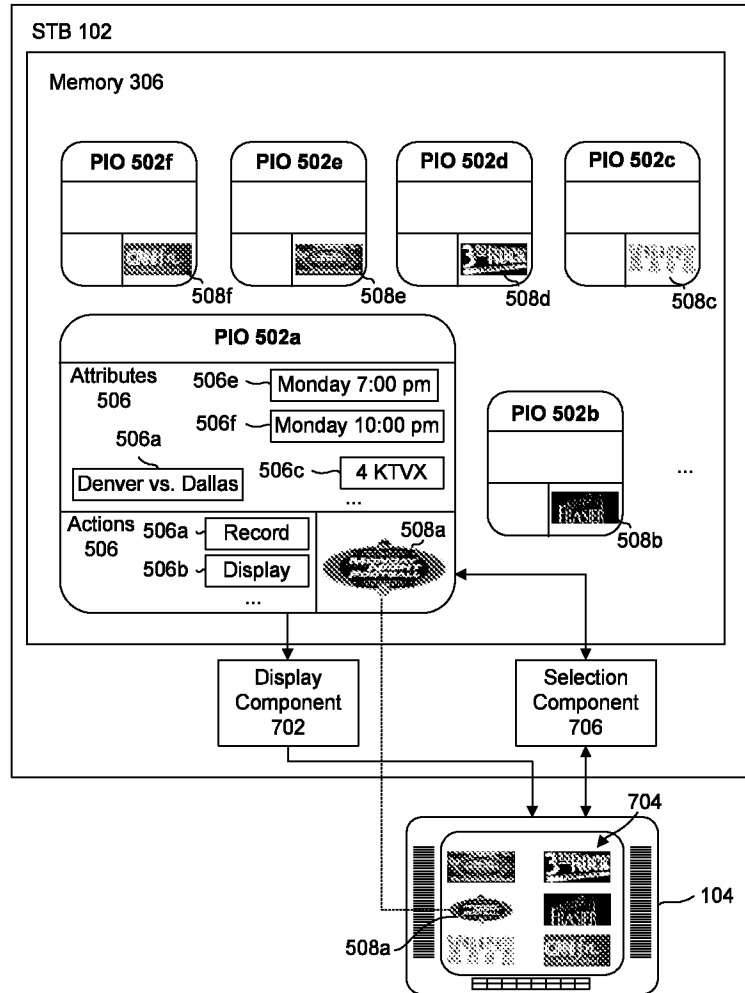
As further illustrated below in FIG. 8 of the present application, the claimed invention relates to a non-traditional type of electronic program guide (EPG), which includes plurality of graphical icons representing PIOs stored within the memory 306. To access the actions of a PIO, the user selects one of the graphical icons, which results in the display of a context menu (802). The context menu (802) lists the available user-selectable actions. When the user has selected one of the actions, the system will execute the routine within the PIO associated with the selected action.
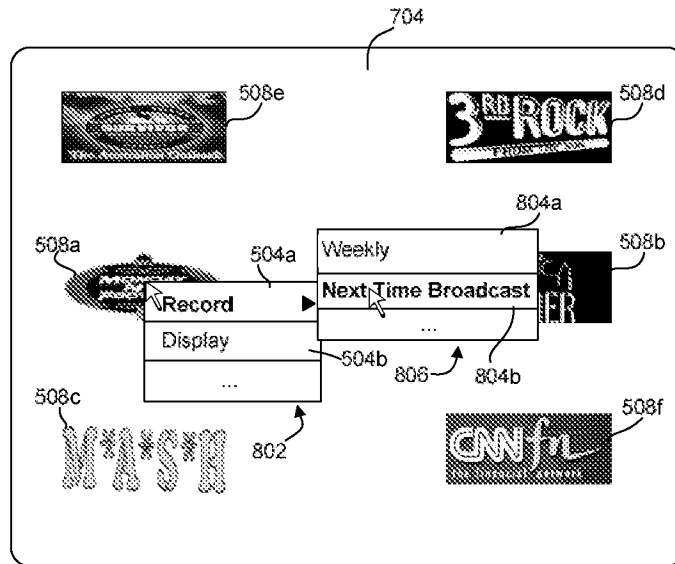
Contrast these claimed features depicted above with conventional approaches in which a text data file comprising program listing data is converted into a grid-based EPG consisting of rows and columns corresponding to channels and time slots, respectively (see FIG. 4, below, illustrating the prior art).

Text Data File 410

Dallas vs. Denver, 2 KUTV, 13:00, 16:00, Football Game
Beatles Countdown, 3 KJZZ, 13:00, 15:00, Music Program
Regis & Kelly, 4 KTVX, 13:00, 14:00, Talk Show
Live with Willie Nelson, 5 KSL, 13:00, 16:00, Music Program
Gun Control Debate, 6 HBO, 13:00, 16:00, Talk Show
Algebra I, 9 KULC, 13:00, 14:00, Educational Program
Trigonometry I, 9 KULC, 14:00, 15:00, Educational Program
Montel Williams, 4 KTVX, 14:00, 15:00, Talk Show
. . .

EPG 402

404

	7:00 PM	8:00 PM	9:00 PM
2 KUTV	Dallas vs. Denver		
3 KJZZ	Beatles Countdown		Front Page
4 KTVX	Sienfield	Whose line..	Drew Carey
5 KSL	Live with Willie Nelson		
6 HBO	Gun Control Debate		
9 KULC	Algebra I	Trig.	Calculus I

20

1.   Bruck's is silent about specific data structures used to implement his UI.

The Office Action appears to be arguing that the various user interfaces depicted, for example, in FIGs. 2, 5-9, and 12 are "web pages," and that these web pages somehow constitute "objects" that "encapsulate" the claimed "attribute data" and "program code." Applicant respectfully submits that Bruck is silent about the specific data structures used in FIGs. 2, 5-9, and 12. The passage of Bruck referred to by the Examiner (*i.e.*, col. 9, lines 32-57) relates, instead, to FIG. 14 and a chat interface.

Assuming, *arguendo*, that FIGs. 2, 5-9, and 12 are "web pages," Applicant respectfully submits that Bruck is silent about whether the displayed UIs are the result of one or many web pages, or whether the recited elements are all contained within the same web page. For example, FIG. 12 of Bruck (reproduced below) illustrates various buttons (record, remind, done), which the Examiner is apparently associating with the user-selectable actions. However, the program code for those actions could easily be included in separate programs (on different computers) or in separate web pages, as shown in the hypothetical figure to the right of FIG. 12.

FIG. 12

21
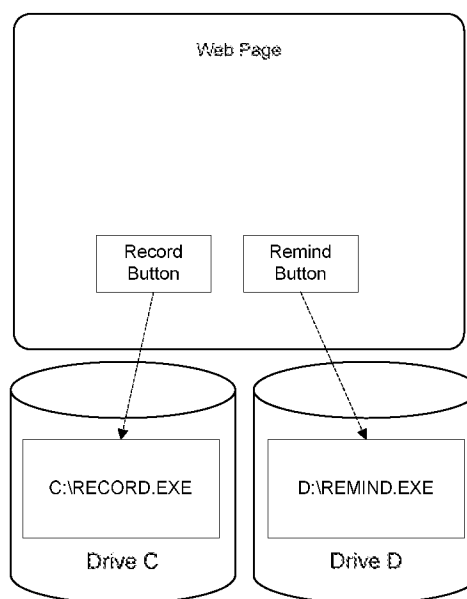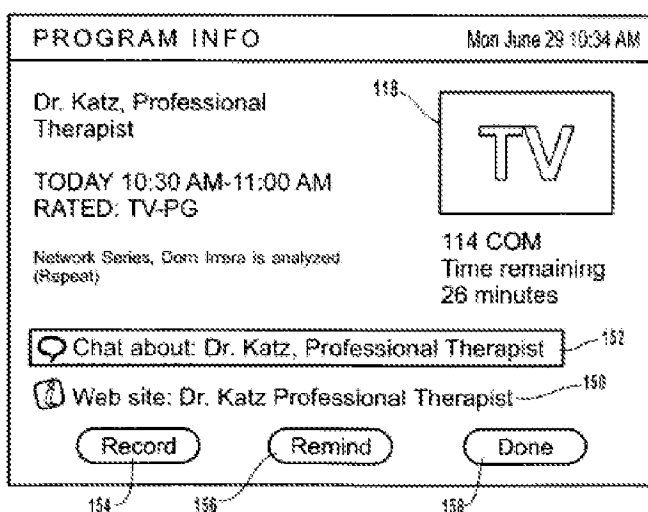
Thus, Bruck's user interface (*e.g.*, FIG. 12) provides virtually no information about the underlined data structures being employed or where the recited data/program code are stored.

2. Bruck does not teach or suggest that PIOs include *routines* for implementing associated actions.

The Office Action appears to be arguing that the claimed "program code" for carrying out the user-selectable actions comprises "HTML" code. Office Action at page 4. However, as noted above, there is no way from simply looking the user interfaces to determine where the program code for carrying out the actions is located. For example, Bruck's FIG. 12 (shown above) illustrates buttons, *e.g.*, "record," "remind," etc. If FIG. 12 is a web page, the buttons are most likely implemented as hyperlinks, which call other web pages or execute externally stored scripts. Applicant assumes, therefore, that the Examiner is associating the claimed "program code" with hyperlinks.

Applicant has amended the claims to remove any doubt that the claim is not referring to hyperlinks. As amended, claim 1 recites that the program code stored within the PIO comprises a routine in a machine independent format that is executable in a Java virtual machine. Support for this amendment may be found, for example, in pages 19 and 20, in which the specification teaches that actions may be represented within the PIO as "program code" in a machine-independent format, and that the actions may also be termed "methods" or "functions." The term "method" is understood in the art to mean:

> In object-oriented programming, a procedure that is executed when an object receives a message. A method is really the same as a procedure, function, or routine in

22

procedural programming languages. The only difference is that in object-oriented programming, a method is always associated with a class.
(http://www.webopedia.com/TERM/m/method.html)
(Emphasis added).

The term "function" means:

(1) In programming, a named section of a program that performs a specific task. In this sense, a function is a type of procedure or routine. Some programming languages make a distinction between a function, which returns a value, and a procedure, which performs some operation but does not return a value.
(http://www.webopedia.com/TERM/f/function.html)
(Emphasis added).

While it may be possible to confuse a hyperlink with the claimed "program code," it is not possible to confuse hyperlink with a "routine" or "procedure," which are generally associated with a plurality of instructions, e.g., "[a]n ordered set of tasks for performing some action." (http://www.webopedia.com/TERM/p/procedure.html). A hyperlink would not normally be referred to as a "routine" by a person having ordinary skill in the art.

Applicant respectfully submits that Bruck is silent about including routines within an object (PIO) to carry out one or more user-selectable actions in connection with a television program. Indeed, Bruck says nothing about "objects" or "methods."

3. Bruck does not teach or suggest that actions are encapsulated within a PIO as routines in a **machine independent format** that are executable in a Java virtual machine.

Applicant respectfully submits that the Office Action did not point to a "virtual machine" in Bruck, as recited in claims 54 and 55 (now incorporated into claim 1), much less a "Java virtual machine," as now claimed. A virtual machine is normally defined as "a self-contained operating environment that behaves as if it is a separate computer."

23

(http://www.webopedia.com/TERM/v/virtual_machine.html). For example, Java applets run in a Java virtual machine (VM) that has no access to the host operating system. Thus, a Java application will run the same in any Java VM, regardless of the hardware and software underlying the system. Because the VM has no contact with the operating system, there is little possibility of a Java program damaging other files or applications.

Applicant notes that Bruck is also silent about the claimed PIOs being implemented as JavaBean objects, as recited in 15, 31, and 51.

4. **Bruck not teach or suggest storing graphical data for display as an icon to _visually represent_ the PIO within a graphical user interface, nor does Bruck teach displaying the icons corresponding to a plurality of PIOs within a graphical user interface _other than a grid-based EPG_ comprising rows corresponding to television channels.**

With regard to claim 6 (now canceled and substantially incorporated into claim 1), the Office Action argued Bruck teaches "displaying the visual indicator in a graphical user interface (i.e. displaying interactive icons, logo, etc. in an interactive user interface – see figures, 6-12)." Office Action at page 6. However, Bruck fails to teach the claim limitations for at least two reasons.

First, Bruck does not disclose or suggest that the graphical data is stored in the same data structure as the attribute data and program code (routines). The logos and the like illustrated in FIGs. 6-12 are almost certainly GIF or JPEG images stored externally from the web page and rendered at the time the web page is displayed. They are not likely encapsulated within the PIO, as claimed.

Second, Bruck does not teach or suggest that the icons are used to "visually represent" the PIO within a graphical user interface. The term "represent" is generally understood to mean "to serve to express, designate, stand for, or denote, as a word,

24

symbol, or the like does; symbolize." *Dictionary.com Unabridged (v 1.1)*. Random

House, Inc. 31 May. 2007. < http://dictionary.reference.com/browse/represent>.

The "NBC" icon of Bruck's FIG. 6 does not "stand for" or "symbolize" a PIO within

a graphical user interface. It is merely one of a number of icons displayed in a "Chat

Transcript" window. By contrast, FIG. 7 of the present application illustrates how a user

interface (704) includes a plurality of icons ***representing*** the PIOs in memory.
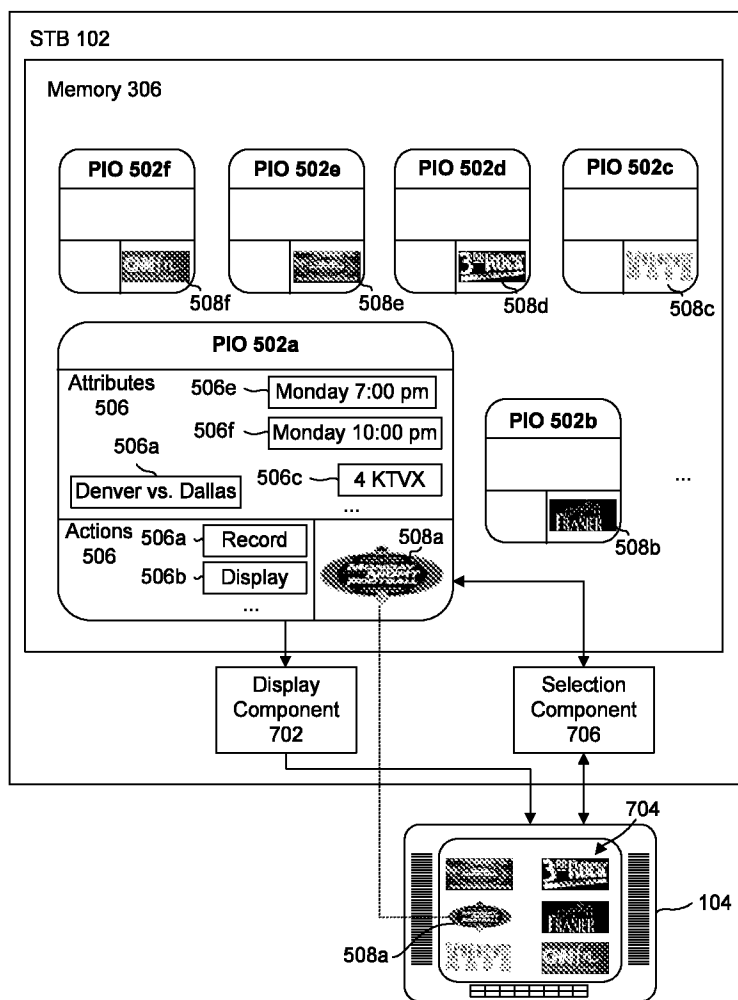
As claimed, the user interface (704) is a non-traditional type of electronic program guide

(EPG), which does not include rows and columns corresponding to channels and time

slots (as illustrated in FIG. 4 of the present application).

Bruck is silent about displaying a plurality of icons representing a plurality of different PIOs within the same graphical user interface. Accordingly, Bruck does not anticipate or render obvious the claimed invention.

5.   Bruck does not teach or suggest the recited method for initiating PIO actions.

As amended, claim 1 recites:

> displaying the icons representing the plurality of PIOs in the GUI;
> in response to a user picking a icon of a selected PIO within the GUI, displaying a context menu listing the one or more actions associated with the selected PIO, at least one of the actions for displaying supplemental information related to the associated television program;
> in response to the action for displaying supplemental information being selected from the context menu, using the at least one attribute comprising the link to retrieve the supplemental information referenced by the link; and
> displaying the supplemental information in the GUI.

These claimed features are illustrated in FIG. 8 of the present application (reproduced below), in which a user selects one of iconic representations of PIOs and is presented with a context menu (802) listing the available actions for selected PIO.

26
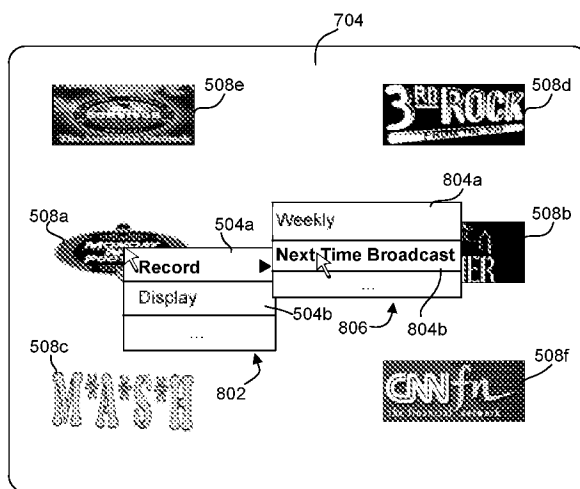
Applicant respectfully submits that claimed process for displaying supplemental content related to a television program is not taught or suggested by Bruck. Bruck doesn't display a plurality of icons representing different PIOs. Bruck does not allow a user to select a particular PIO via a displayed icon. Bruck does display, *in response to one of the icons being selected*, a context menu listing the actions available for the associated PIO, one of the actions being configured to display supplemental content.

Conclusion

In view of the foregoing, claim 1 is believed to be patentably distinct. Claims 37 and 52 have been amended to include similar limitations and are likewise believed to be patentably distinct for at least the same reasons. All other claims depend directly or indirectly from one of the foregoing claims and are therefore patentable by virtue of that dependency. A Notice of Allowance is therefore respectfully requested. The Examiner is encouraged to contact the undersigned at the telephone number provided below for a quick resolution of any remaining issues.

Respectfully submitted,

Digeo, Inc.

By /Kory D. Christensen/
Kory D. Christensen
Registration No. 43,548

STOEL RIVES LLP
One Utah Center Suite 1100
201 S Main Street
Salt Lake City, UT 84111-4904
Telephone: (801) 328-3131
Facsimile: (801) 578-6999